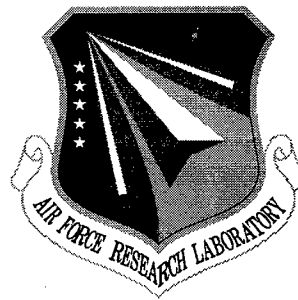AFRL-IF-RS-TR-1999-226
Final Technical Report
October 1999

SYSTEM SUPPORT FOR DISTRIBUTED SUPERCOMPUTING ON A NETWORK OF WORKSTATIONS (NOW)

University of California, Berkeley

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

19991222 073
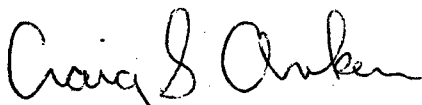
**AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK**

DTIC QUALITY INSPECTED 4

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-1999-226 has been reviewed and is approved for publication.

APPROVED: *Craig S Anken*

CRAIG S. ANKEN
Project Engineer

FOR THE DIRECTOR: *Northrup Fowler*

NORTHRUP FOWLER, III, Technical Advisor
Information Technology Division
Information Directorate

SYSTEM SUPPORT FOR DISTRIBUTED SUPERCOMPUTING ON A NETWORK OF WORKSTATIONS (NOW)

David E. Culler
Thomas E. Anderson
David A. Patterson

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED	
	October 1999	Final	Mar 95 - Sep 98

4. TITLE AND SUBTITLE

SYSTEM SUPPORT FOR DISTRIBUTED SUPERCOMPUTING ON A NETWORK OF WORKSTATIONS (NOW)

5. FUNDING NUMBERS

C   - F30602-95-C-0014
PE - 62310E
PR - C137
TA - 00
WU - 01

6. AUTHOR(S)

David E. Culler, Thomas E. Anderson, and David Patterson

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

University of California, Berkeley
Department of Computer Science
627 Soda Hall, Computer Science Division #1776
University of California, Berkeley, CA 94720-1776

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/IFTB
525 Brooks Road
Rome NY 13441-4505

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

AFRL-IF-RS-TR-1999-226

11. SUPPLEMENTARY NOTES

Air Force Research Laboratory Project Engineer: Craig S. Anken/IFTB/(315) 330-4833

12a. DISTRIBUTION AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The goal of the NOW project was to explore and demonstrate a fundamental change in approach to the design and construction of large-scale computing systems. This was motivated by the desire to deploy powerful systems very rapidly and to scale them incrementally, as is required to fully utilize commercial technologies that are advancing at a high rate, to meet new service demands that are increasing "on internet time," and to address emergency or military situations. The key enabling technology for the project was the emergence of scalable, low-latency, high-bandwidth VLSI switches, pioneered in massively parallel processors and transferred into system area network (SAN) configurations. With SAN technology, it became feasible to construct powerful, integrated systems by literally plugging together many state-of-the-art commercial workstations or PCs to form a high performance cluster. The project demonstrated the design approach, the solution to core challenges, and novel design opportunities by building and utilizing a cluster of over one hundred Ultrasparc workstations interconnected by a multigigabyte Myricom network.

14. SUBJECT TERMS			15. NUMBER OF PAGES
Networks, Workstations, PC, Processing			44
			16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

CONTENTS

Executive Summary

The goal of the NOW project was to explore and demonstrate a fundamental change in approach to the design and construction of large-scale computing systems. This was motivated by the desire to deploy powerful systems very rapidly and to scale them incrementally, as is required to fully utilize commercial technologies that are advancing at a high rate, to meet new service demands that are increasing "on internet time," and to address emergency or military situations. The key enabling technology for the project was the emergence of scalable, low-latency, high-bandwidth VLSI switches, pioneered in massively parallel processors and transferred into system area network (SAN) configurations. With SAN technology, it became feasible to construct powerful, integrated systems by literally plugging together many state-of-the-art commercial workstations or PCs to form a high-performance cluster. However, to realize the potential of SAN-based cluster design, critical limitations of existing systems had to be overcome. Particularly, the costly communication stack forming the software interface between the user application and the network had to be essentially eliminated, while preserving the ability to share communication resources effectively. In addition, the operating system functionality associated with an individual node had to be made available cluster-wide. Overcoming these challenges enabled many novel design concepts, such as file and virtual memory systems that utilize remote memories in preference over disks, schedulers that coordinate actions implicitly, massively scalable I/O, and scalable active services. The project demonstrated the design approach, the solution to core challenges, and novel design opportunities by building and utilizing a cluster of over one hundred Ultrasparc workstations interconnected by a multigigabyte Myricom network.

This prototype cluster enabled many further investigations; it was basis for Inktomi (now the world's largest search engine), set the world records for throughput and response time on the disk-to-disk database sort benchmark and held them for two years, broke the RSA key challenge, put clusters on the Linpack Top-500 list, hosted the transcoding proxy service (BARWAN Transcend) and the media gateway service (MASH), provided the simulation engine for processor-in-memory (IRAM) and configurable architectures (BRASS), enabled real-time video effects processing and allowed massive image processing in digital libraries. Technology from the project has transferred to industry in many forms, including the recent Intel/Microsoft/Compaq Virtual Interface Architecture (VIA).

The first major application breakthrough on NOW was Inktomi, which debuted on a segment of the NOW cluster in summer 1995 as the first search engine to offer fast response time and a large search set. It utilized the capacity and parallel transfer of cluster I/O to support a very large database, while using the fast communication layer to fully parallelize each query for fast response. The approach and the underlying communication technology transferred to the company, which powers many of the world's leading search sites, including HotBot, NBC's Snap!, Yahoo!, and the Disney Internet Guide (DIG) for children and families. Additional advances came through the development of active infrastructure services, in collaboration with the MASH and BARWAN projects, which place services into the infrastructure that adapt content and access to the needs of a large number of potentially limited clients. For example, the Transcend proxy provides not only caching, but actively renders pages and reprocesses images into a form where they can be transferred quickly over a low-speed link and presented on a small Palm Pilot. The Media Gateway participates in an mbone session while transcoding the video

stream to match the limited connection of its client. Each of these resides permanently in a portion of NOW, but grow out to utilize cluster resources as demand increases. In addition, NOW has been made available to the national computation science and computer science community as an experimental resource with the National Partnership for Advanced Computing Infrastructure (NPACI).

An essential technology developed by NOW was fast, general purpose user-level communication using virtual networks. This built upon the Berkeley work on Active Messages, which established the paradigm for fast user-level messaging on MPPs, but provides the generality of a LAN within the same overhead budget of a microsecond or two. The central idea is to provide user applications and OS subsystems with the illusion of direct, dedicated access to the network, while binding virtual networks to physical resources in a protected fashion on demand. Communication is integrated with virtual memory management, so that message arrival or processor access to an object the represents an endpoint of a virtual network causes the bond to be established. An intelligent network interface is then able to manage communication flows between the network and the most active subset of the endpoints without operating system intervention.

Thus, several applications can each have specific, dedicated communication run-times, while a parallel file system has its own virtual network, and the kernel provides IP communication within another. Meanwhile, the communication subsystem can probe the physical network, detect changes, and perform diagnostics or reconfiguration. The VIA standard builds directly on this work, along with that of the Unet and Shrimp projects, but provides an endpoint that is similar to traditional LAN interfaces. Active Messages, in the general form provided in NOW, are also deeply related to the Active Networks efforts.

NOW advanced the concept of a cluster-wide operating systems layer that organized the large pool of resources into a federation. Unlike previous single system image work, Glunix sought to isolate the impact on the kernel while providing substantial capabilities through user-level mechanisms. One of the most novel design aspects was coordinating the scheduling of applications across many nodes, only when necessary and while preserving the autonomy and robustness of local schedulers. Coordinated scheduling is most important when an application communicates intensively or has strong dependences among its components. Simple mechanisms within the user-level runtime allow these situations to be detected and the local schedulers to be driven toward coordination by adjusting the spin-block behavior of the application. A unique capability of clusters is the independent I/O system of every node, which provides tremendous aggregate bandwidth, combined with the communication capacity that makes the bandwidth accessible. The project investigated several novel subsystem designs, including the xFS file system that utilized cooperative caching to maximize the utility of the aggregate client caches, network RAID for bandwidth and availability, network virtual memory, and highly tuned flexible striping. Building on the high-performance sorting work, a very general notion was developed of parallel I/O "rivers" which are used to plumb together scalable, robust data intensive applications.

The NOW project is an excellent example of the positive cycle of building and using experimental systems. It built several prototypes, utilizing a variety of potential system area

2

network technologies (including repackaged MPP networks, Hewlett Packard's experimental Medusa interface, Myrinet predecessors, and numerous commercial ATM options) and a variety of operating system substrates, while advancing the larger system design. By bringing prototypes to an operational state and supporting applications from a variety of other research projects, an understanding of new requirements and novel opportunities emerged, which fed back into the system design. This process was carried out with strong federal support and widespread industrial collaboration in a very open forum, allowing rapid technology transfer in many directions. Today, cluster technology is widespread in research and commercial settings.

All of the NOW software and publications are freely available at http://now.cs.berkeley.edu.

Review of Proposed Work

We proposed to build system support for using a network of workstations (NOW) as a distributed supercomputer, on a building-wide scale. Our goal was to demonstrate a practical 100 processor system that delivers at the same time (1) better cost-performance for parallel applications than a massively parallel processing architecture (MPP) and (2) better performance for sequential applications than an individual workstation. This goal required combining elements of workstation and MPP technology into a single system. Because of volume production, commercial workstations offer much better cost-performance than the individual nodes of MPPs; in addition, we forecasted the move towards cheap, high-bandwidth, switched local area networks. But in order to realize the potential of NOW for parallel processing, we needed to move two MPP technologies into the workstation community: low latency networks and global system software that treats a collection of processors as if they were a single machine.

Our approach was a coordinated attack on the crucial pieces of system support for a NOW: network interface hardware, communications protocol software, local kernel modifications, and a global system layer. Instead of building everything from scratch, we proposed to use off-the-shelf workstation hardware, workstation operating systems on each node, and local area network switches. We saw a number of challenges to making a NOW practical for both sequential and parallel computing in a multiprogrammed environment; and believed that solutions to these challenges would require re-thinking the traditional division of labor between hardware and software, application and operating system, compiler and communication software. From the outset, a number of issues differentiated the NOW project from other "cluster computing" efforts: scale (our target was 100s of workstations, not 10s), communication performance (we sought to provide user-to-user communication at under 10 microseconds, not 100s or 1000s of microseconds), design for technology transfer (we were working with workstation, MPP, and network vendors, building on top of artifacts they are already producing), the focus on practical concerns with a NOW (preserving interactive performance, providing high-performance I/O), coordinated hardware/software research attack, and adaptation to the requirements of large scale parallel programs.

The starting point for our work was Active Messages, a new communication architecture that had become a de facto standard for low-latency communication in MPPs. This abstraction provides a simple, flexible user-to-user communication primitive, with little of the buffering, storage management, and scheduling overhead of traditional message passing layers. It also provides a basis for global address programming models. We intended to build network interface hardware and software to support Active Messages on a general-purpose NOW, while keeping communication performance competitive with a dedicated MPP. To avoid the layers of protocol software currently needed on workstations, we will run each parallel program within a network protection domain, called a network process. Messages were to be directly user-to-user, with protection enforced by simple checks in the network interface card, instead of by software. In addition, we will develop extremely light-weight flow control and error handling techniques, because the network will be a shared resource and only nearly reliable, instead of dedicated and perfectly reliable as in MPPs. The goal: in the frequent case, communication within a parallel program will need little overhead and no operating system intervention.

Volume production had not only led to better cost-performance of workstation hardware relative to MPP hardware, it also meant that workstation software has better reliability and functionality than that found on an individual node of an MPP. Yet fast communication requires operating system support, so that the pieces of the parallel program are co-scheduled at the same time on all of its processors. We built a new global system layer, GLUnix, to provide global system services, including co-scheduling, resource allocation, and parallel file operations. Instead of starting from scratch, we constructed GLUnix as a layer on top of existing standard commercial operating systems. GLUnix requires the development of a narrow, backwardly compatible extension of the Unix interface to allow global control over local scheduling. The GLUnix layer will provide a parallel file system in a similar manner, using striping and log structuring techniques to obtain very high file bandwidth and very high availability across the network.

We also proposed to develop solutions to practical problems limiting the use of a NOW for general purpose parallel and sequential computing. One key is to preserve good interactive response time, so that the cluster is seen as better than a single workstation, even for those users with no parallel jobs to run. Another key was that the global layer software must be fault-tolerant—individual machines in a NOW can fail or be replaced without affecting programs running on other nodes. We must also provide automated system management; the industry standard of a system manager per 12 workstations is not scalable in a NOW.

What differentiated NOW from MPP efforts was the use of complete state-of-the-art commercial systems—not just the processor, but the motherboard, the disks, and the operating system as well. This approach will reduce the lag time for uniprocessor technology to be incorporated into a parallel system, improving cost effectiveness. This has a profound impact on the research approach: one must adopt a "higher order" systems perspective and work within constraints imposed by existing hardware and software components, rather than build from scratch.

The project was to be carried out in close collaboration with workstation, MPP, and network vendors, providing immediate avenues for direct technology transfer. Indeed, the total industrial participation in the project exceeded $3 million, compared to the DARPA contribution of $3.6 million.

We sought to demonstrate that a network of workstations (NOW) could be used as the everyday computing infrastructure of science and engineering, for both parallel and sequential computing. A NOW as infrastructure would leverage a number of existing building blocks–state-of-the-art workstations complete with local disks, widely-used commercial operating systems, commercial application software, and a high-bandwidth, switched local area network. Our research would show that a small amount of mortar can glue these building blocks into a practical computing system on a building-wide scale of 100s of processors,· providing benefits for both parallel computing users and interactive sequential users. Our glue would be a global layer operating system built on top of a slightly modified existing commercial OS, along with network interface hardware and software designed for low latency communication over a local area network.

In making a NOW practical for everyday computing, we sought to demonstrate solutions to a number of technical problems:

Operating Systems:

- We would define a parallel network process abstraction. A network process includes a local process on each of many workstations, a communication protection domain encompassing these local processes, and co-scheduling of the local processes. This network process abstraction is crucial to achieving high performance communication within a parallel program, by eliminating the need for protection checks, buffering, and context switches on each message sent over the network. We would show how a small set of backwardly compatible changes can be made to industry standard operating systems to make them "parallel ready" - that is, able to support the network process abstraction.

- We would devise a structuring methodology for the global layer operating system to enable it to be highly available and resilient to node failures. Programs running on unrelated nodes will continue to run unaffected even when one workstation crashes. We will also show how to support incremental changes in hardware and software configuration, without rebooting the entire cluster.

- We would develop a resource allocation policy to maintain fast interactive performance for sequential programs. As we recruit machines, processing cycles, memory, disk bandwidth for running a set of parallel programs, sequential users will not be able to notice the difference between a dedicated workstation and a NOW cluster--except that by using a NOW, a sequential program will be able to utilize the entire I/O bandwidth of the cluster, if needed. Preserving interactive performance involves migrating parallel processes off a previously idle node and returning the memory of the node to its original state, all done automatically in the gap between when a user steps in their office (setting off a motion detector), and when they sit down to type at the keyboard. We will also investigate harvesting computing resources from underutilized workstations, by reserving sufficient idle capacity to absorb increases in interactive demand.

- We would show how to build a parallel network file system constructed out of local file systems on many workstations. This will provide high bandwidth I/O for parallel programs. Capabilities for structuring the layout of parallel files will be provided to allow for optimization of local accesses in a manner analogous to data layout in High Performance Fortran.

Network Interface Hardware and Software:

- We would show how extensions of Active Messages can provide highly efficient communication within a network process in a general purpose parallel computing system distributed over a building. In particular, we will show how to integrate fast communication, which demands essentially direct network access, with a full function operating system. This communication abstraction will provide the basis for a wide variety of parallel programming models on a NOW.

- We would show how to provide low overhead, high bandwidth delivery of Active Messages on stock workstations and demonstrate a communication architecture that scales across the range of access points available in commercial workstations, from simple I/O busses on the low-end to fast, cache-coherent memory busses on the high end. We will demonstrate the design in a network interface card.

6

- We would show how simple hardware assists in the network interface card can (1) support zero-overhead dynamic protection checks on user-to-user messages, in the common case, as required by the network process abstraction, (2) allow communication software to be optimized for reliable message delivery and still handle lost messages as the exceptional case, and (3) provide simple and effective flow control for high-quality networks.

In summary, the NOW project was to be an example of "higher order" systems research, whereby we build on top of existing hardware and software systems rather than from scratch. Examples of higher order systems research include the global layer operating system built on top of local operating systems, the parallel network file system built on top of local file systems, and a high performance network using custom network interfaces to connect off-the-shelf workstations via a standard switch. It would help DoD both in the cost-performance of computing and in systems development time. NOW would not only lower the cost of parallel computing, accelerating the use of parallelism and therefore reducing the time to get answers to important questions, but NOW would also demonstrate that effective systems can be build on top of commercial hardware and software platforms as opposed to starting from scratch, as is traditional in DoD systems.

The overall rationale and approach of the project is described in:
- Thomas E. Anderson, David E. Culler, David A. Patterson, and the NOW Team, "A Case for Networks of Workstations: NOW." *IEEE Micro*, February 1995. (Available as Abstract and PostScript)

Summary of Accomplishments

We constructed a series NOW prototypes to explore key design issues and technologies. These delineate the major phases of the project.

Preliminary Prototype

The first NOW-0 prototype consisted of four HP PA/735 workstations interconnected by an FDDI network through an experimental network interface (Medusa) on the proprietary graphics bus, shown below. Using this platform, we developed an initial version of Glunix over HPUX and developed the first user-level Active Networks implementation in a general purpose environment, HPAM. The HPAM work was presented in the Hot Interconnects Symposium and served as a driver for the Generic Active Messages work.

Detail appears in
* Richard P. Martin, "HPAM: An Active Message Layer for a Network of HP workstations," *Hot Interconnects*, 1994. (Available as PostScript).

NOW-0 Prototype: four HP P`A/735 with an experimental FDDI network on the graphics bus.
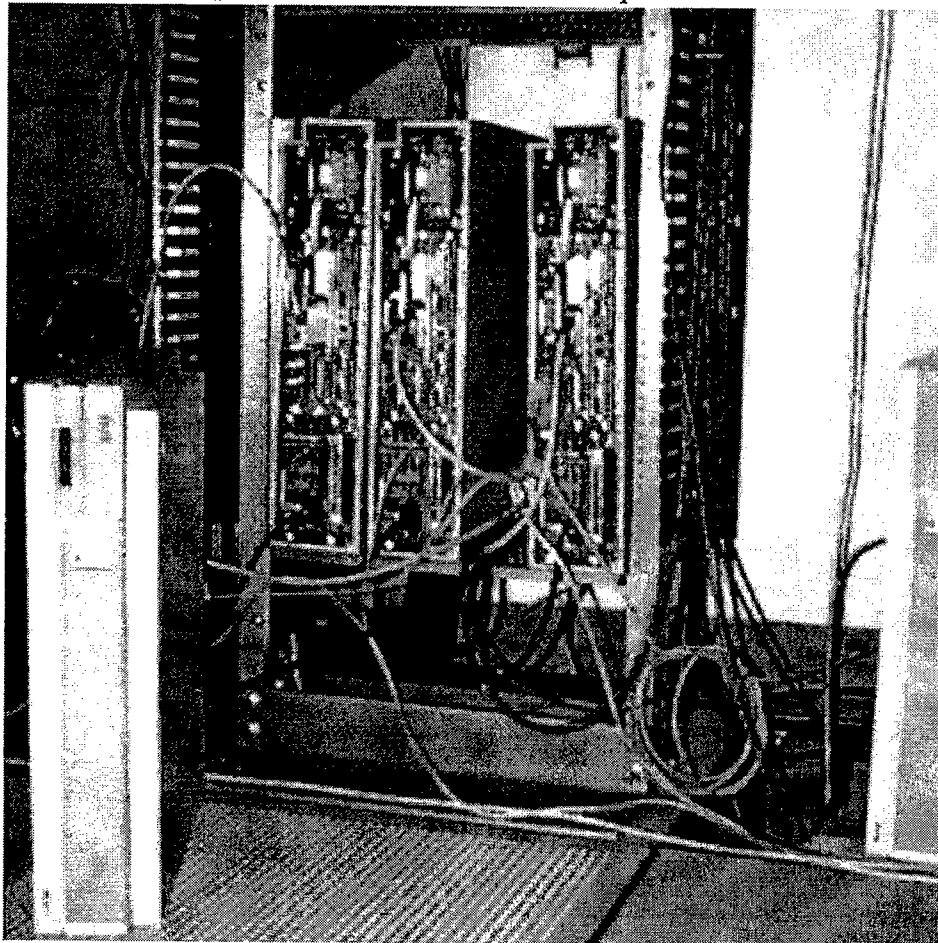
Technology Exploration Prototype

The second prototype, called NOW-1, consisted of 32 SparcStation 10s and 20s connected by Ethernet, ATM, and Myrinet (shown below). NOW-1 was up in late Spring 1994 with ATM. This provided a basis for extensive analysis of emerging high-bandwidth, low-latency interconnect technologies. We built a testbed with several different ATM networks and eventually built out a BayNetworks ATM configuration for the entire cluster; this was working in May 1994. We also experimented with the Atomic network and then built out a Myrinet network for the entire cluster. At the time, this was the largest Myrinet configuration in existence and was operational by May 1995.

NOW-1 Prototype: 32 Sun SparcStation 10s and 20s with numerous candidate cluster networks, including ATM and Myrinet
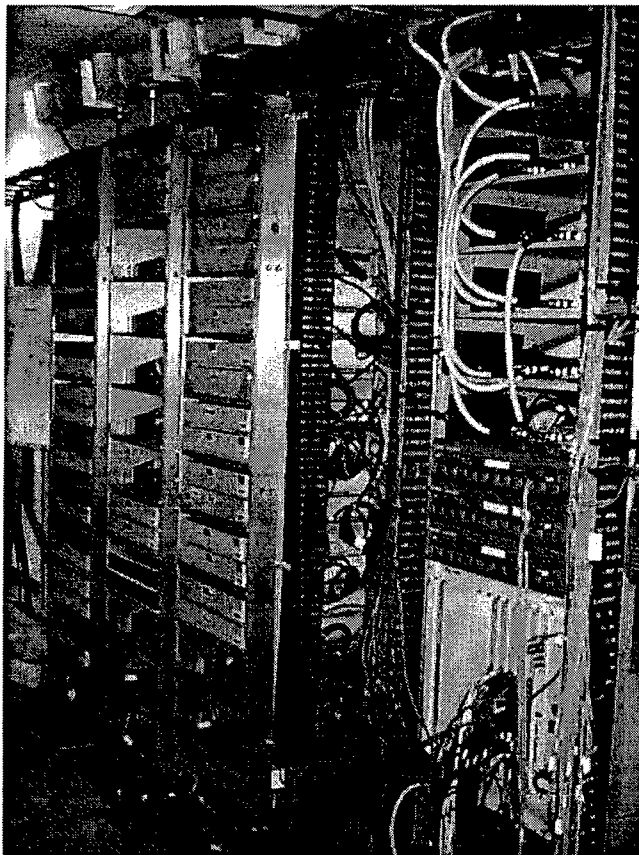
Fast User-level Communication

We built Active Message implementations directly for the Sun SAHI ATM network interface as well as for the Myricom Lanai 2.3 network interface. This investigation made clear that it was

possible to design a generic Active Message layer for a wide class of network interface architectures with little loss of performance relative to APIs that were tailored to the specific platform. This examination was codified in the following reports:

- David Culler, Kim Keeton, Lok Tim Liu, Alan Mainwaring, Rich Martin, Steve Rodrigues, Kristin Wright, Chad Yoshikawa. "The Generic Active Message Interface Specification." *White Paper*, 1994. (Available as: PostScript)

- Eric Brewer, Frederic Chong, Lok Liu, John Kubiatowicz, Shamik Sharma, "Remote Queues: Exposing Network Queues for Atomicity and Optimization," *Proceedings of SPAA*, 1995. (Available as: PostScript)

Based on this work, we conducted the first broad spectrum analysis of high performance network interface architectures. This involved developing high performance implementations of Generic Active Messages (GAM) for key points in the design spectrum: a simple NI directly on the memory bus (Thinking Machines CM-5), an NI supported by a full microprocessor (Intel Paragon), an NI with an embedded processor on the memory bus (Meiko CS-2), a simple NI on the graphics bus (HP Medusa), a simple NI on an I/O bus (Sun ATM), and an NI with embedded processor on an I/O bus (Myricom Lanai). We developed a sophisticated microbenchmarking tool to isolate the basic parameters of the LogP model - latency, overhead, and bandwidth - using black-box empirical techniques. We achieved our goal of 10 us user-to-user communication time.

Details of the implementations, benchmarking techniques and architectural evaluation are provided in the following:
- Lok Liu, David Culler, "Evaluation of the Intel Paragon on Active Message Communication," *Proceedings of Intel Supercomputer Users Group Conference*, 1995. (Available as: PostScript)

- Lok Liu, David Culler, "Measurement of Active Message Performance on the CM-5," *Technical Report*, 1994. (Available as: PostScript)

- Lok Tin Liu, Alan Mainwaring, Chad Yoshikawa, "Building TCP/IP Active Messages," *White Paper*, 1994. (Available as: PostScript)

- Krishnamurthy, K. Schauser, C. Scheiman, R. Wang, D. Culler, and K. Yelick, "Evaluation of Architectural Support for Global Address-Based Communication in Large Scale Parallel Machines" *Architectural Support for Programming Languages and Operating Systems*, 1996. (Available as: PostScript)

- David Culler, Lok Tin Liu, Richard Martin, Chad Yoshikawa, "LogP Performance Assessment of Fast Network Interfaces," *IEEE Micro*, 1996. (Available as: PostScript)

- Kimberly Keeton, Thomas Anderson, David Patterson, "LogP Quantified: The Case for Low-Overhead Local Area Networks," *Hot Interconnects III: A Symposium on High Performance Interconnects*, 1995. (Available as: PostScript)

Global Operating Systems Layer

The NOW-1 Prototype provided the basis for the first round of investigations into the design requirements of the global operating systems layer. Although there had been numerous studies showing the ample idle cycles in LAN environments and numerous studies on scheduling parallel workloads, there was little quantitative understanding of how production parallel workloads and interactive workloads would mix. We studied this problem by collecting traces from the CM-5 at Los Alamos National Laboratory and workstations in various university projects and combining them in trace driven simulation. The results were encouraging in showing a surprising degree of compatibility in the two workloads, in terms of CPU patterns, working set sizes, program duration and frequency of migration. Details are in the following:

- Remzi Arpaci, Andrea Dusseau, Amin Vahdat, Lok Liu, Thomas Anderson, David Patterson, "The Interaction of Parallel and Sequential Workloads on a Network of Workstations," *SIGMETRICS '95*, 1995. (Available as: PostScript)

We also explored techniques for building various forms of a signal system image over complete operating systems, as described in the following. We built an initial release of the Glunix global operating system layer for everyday use.

- Amin Vahdat, Douglas Ghormley, Thomas Anderson, "Efficient, Portable, and Robust Extension of Operating System Functionality," *UC Berkeley Technical Report CS-94-842*, 1994. (Available as: PostScript)

- Kim Keeton, Steve Rodrigues, Drew Roselli, "Previous Work in Distributed Operating Systems," *White Paper*, 1995. (Available as: PostScript)

Scalable I/O

The presence of a high performance cluster, low-overhead high-speed networking, and a global system layer allowed us to explore to opportunities for novel approaches to the design of key subsystems. Two of particular importance were the file system and the virtual memory system. We investigated, designed, prototyped and evaluated xFS file system which took the view that disk storage was striped across nodes in the cluster (exploiting the network link bandwidth and the aggregate disk bandwidth), that all nodes provided a large cooperative cache (exploiting the low-latency network to support the protocol and the aggregate memory capacity in commodity nodes), and metadata could be spread over the cluster and replicated for reliability. With a fast network, remote memories are far more accessible than even local disk. Detail of this work is presented in the following:

- Randolph Wang, Thomas Anderson, "xFS: A Wide Area Mass Storage File System," *White Paper*, 1993. (Available as: PostScript)

A key question is how to manage the collection of caches across the cluster. Cooperative management can increase the overall cache coverage with simple mechanisms. When a block is

first cast out, it takes refuge in memory of another node. After a few such chances, it is finally cast to disk. While reducing miss rate from the collective caches, this has the elegant property that a busy node tends to behave like a client while and idle node tends to behave like a server. Detail is provided in the following.

- Michael Dahlin, Randolph Wang, Thomas Anderson, David Patterson, "Cooperative Caching: Using Remote Client Memory to Improve File System Performance," *OSDI 1*, 1994. (Available as: PostScript)

- Michael Dahlin, Clifford Mather, Randolph Wang, Thomas Anderson, David Patterson, "A Quantitative Analysis of Cache Policies for Scalable Network File Systems," *SIGMETRICS '94*, 1994. (Available as: PostScript)

Extending the basic concept of cooperative cache, we also explored and implemented systems that utilize the collective capacity of the cluster memories to support virtual memory. Rather than paging to local disk, page into under-utilized remote memories. This captured the opportunity provided by a complete operating system on every node, with a complete I/O system, in addition to fast networking. It revealed, however, that current VM systems are tuned to slow peripherals. They use complex and costly algorithms to avoid I/O transfers, whereas with network virtual memory the optimal design point would be a very fast trap path, even with more page transfers. Detail is in the following.

- Alan Mainwaring, Chad Yoshikawa, Kristin Wright, "Network RAM," *White Paper*, 1994. (Available as: PostScript)

This technology exploration prototype phase included extensive analysis of the node hardware and operating system requirements, as well as the networking requirements. We were able to demonstrate clearly that the Myrinet technology was superior for this application to ATM, which was anticipated as the commercial breakthrough. However, our approach of pushing scale and completely replacing their network interface firmware in order to push performance and generality also revealed design flaws in their switches. We worked very closely with Myricom to move the technology forward. We entered into the mid-level prototype phase with the expectation that the final demonstration prototype would be comprised of a hundred HP PA/8000-based machines. We developed an network interface card that married the proprietary HP GSC+ bus interface used in HP's fiberchannel adapter with Myricom's Lanai. This initial design was made available to HP and Myricom, who later produced product using the same strategy with a different FPGA. We found the OS threads and the segmented VM system of Solaris to be critical, and not present in HPUX 10, while the newly announced UltraSparc had comparable performance, better physical packing density, and lower power loading than the HP J-series, with a standard I/O bus that was heavily targeted by emerging network interfaces.

Final Demonstration Prototype

Our final demonstration prototype consisted of 110 Sun UltraSparc 170s, purchased partly on this DARPA grant, partly on the Titan NSF Research Infrastructure Grant, and partly donated by Sun Microsystems. Each node has two internal disks and 128 MB of memory. Several nodes

have additional disk or memory capacity. These were configured into a very large Myrinet using over 60 8-port switches. When constructed, this was once again the largest Myrinet network and pushed the technology forward. The figure below shows the initial configuration. It was later replaced with a cleaner 16-port rack-mounted switch design and 42 switches, after revealing certain, very subtle design flaws. Pushing the envelope of the technology not only improved system area networks for all cluster efforts, it focused our worked on light-weight techniques for tolerating failures in high performance networks. The entire NOW software architecture was developed and deployed on this cluster and a wide range of applications served as a basis for its evaluation.

NOW Final Demonstration Prototype: 110 Sun UltraSparc 170s
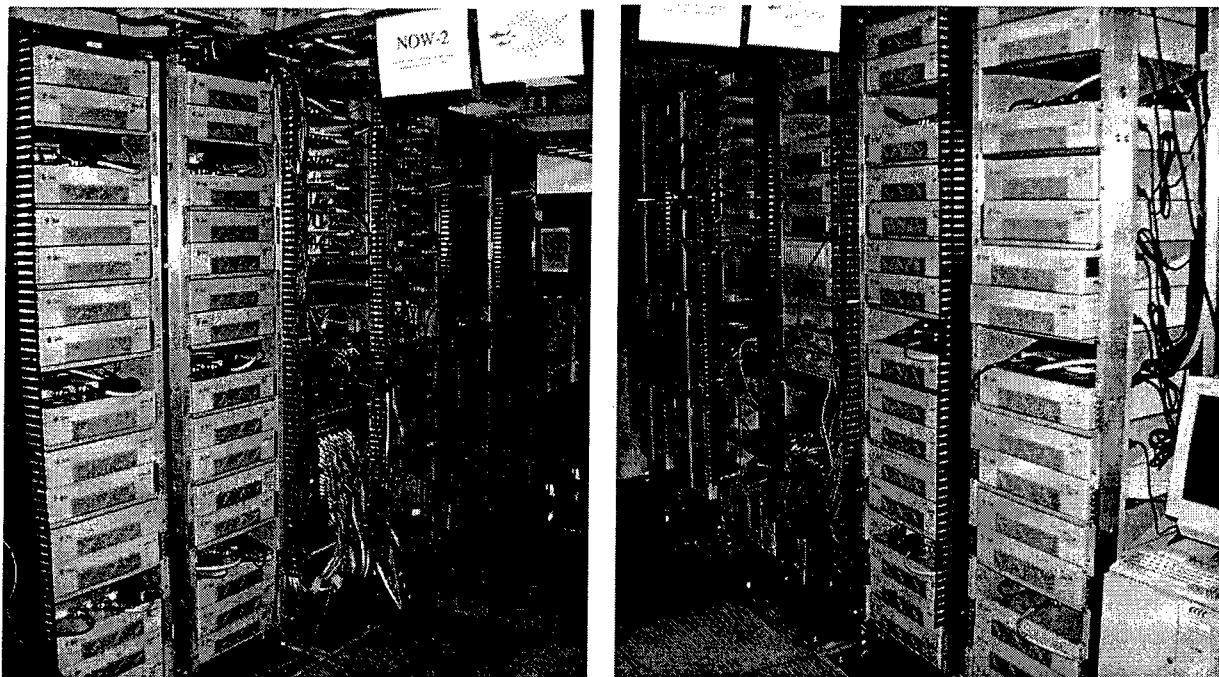
High-Performance Networking

We developed a fast, robust, general purpose Active Message layer as the foundation for high-performance communication in NOW. It extends the traditional Active Message lightweight RPC model with more powerful naming, protection, and error handling, while preserving much of the simplicity of the transport operations. The following document is the specification of the API. A logical attachment of a process to a network is abstracted as endpoint object. Many processes on a node can have one or more endpoints. A collection of such endpoints, spanning one or many processors, forms a virtual network. It is an addressing domain and a protection domain, and provides basic quality of service. The approach is powerful enough to support a very rich set of communication relationships: traditional parallel programs, client/server, parallel

13

clients, parallel servers. Furthermore, to allow programs to handle faults without defensive programming on the critical path, the API provides a novel "return to sender" error model. If an active message cannot be delivered, it invokes an error handler on the sender on the entire message.

In effect, virtual networks provide a horizontal networking abstraction, complementing the traditional vertical stack. For example, each kernel has an endpoint (forming a privileged virtual network) with a complete IP stack on top it, while user MPI libraries and I/O libraries each go directly to their own endpoints. Endpoints are bound to physical networking resources on demand.

- Alan Mainwaring, David Culler, "Active Messages: Organization and Applications Programming Interface," *Technical Document*, 1995. (Available as: PostScript)

With this approach, multiple active flows are presented, unmultiplexed, to the network interface. This provides a number of challenges and opportunities for lightweight protocols for scheduling traffic for multiple endpoints and for low-level error detection and processing. The following provides detail.

- B. Chun, A. Mainwaring, D. Culler, "Virtual Network Transport Protocols for Myrinet," *Proceedings of Hot Interconnects* V (award paper), August 1997. Selected for *IEEE Micro* Special Issue, Jan/Feb 1998, pp. 53-63 (Available as: PostScript)

Furthermore, the active subset of the endpoints should be kept resident in physical communication resources. Thus, the endpoint abstraction is fundamentally like virtual memory, and it is implemented by an elegant integration with the Solaris virtual memory system. This must have the property that full hardware capability is delivered to a single program in the uncontended case, while a fair fraction is delivered when multiple applications are communicating and the aggregate performance is robust even under heavy load. Detail of the solution and benchmark evaluations is provided in the following.

- Alan Mainwaring and David Culler, "Design Challenges of Virtual Networks: Fast, General-Purpose Communication," *SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPOPP)*, Atlanta, 4-6 May 1999. (Available as: PostScript.)

A key property of our design for NOW is that it be self-configuring and adapt to incremental changes, such as nodes or links being added, removed, or failed. Incremental reconfiguration requires handling communication errors, even of the switches were perfect. In addition, the network must configure itself automatically. This is achieved by providing a privileged endpoint that uses absolute routes, rather than virtual network addressing. A system daemon periodically probes the network to discover the physical topology. (Producing a canonical map from a set of such probes is a challenging theoretical problem solved in the following paper.) From this map a set of deadlock free routes are determined and established in the network interface firmware. (The cluster will work with any topology, but some yield higher performance than others.) Most probes have invalid routes, so the mapper daemon heavily exercises the return-to-sender error model. Moreover, probing on a live system may actually introduce transient deadlocks; these are

handled by the routine timeout detection and retry mechanisms within the network interface firmware. Thus, low-level error handling is a key aspect of general purpose use. (The current network map of NOW is generated interactively at http://www.cs.berkeley.edu/~alanm/map.html.)

- Brent Chun, Alan Mainwaring, Saul Schleimer, Daniel Wilkerson, "System Area Network Mapping," *SPAA '97*, Newport, Rhode Island, June 1997. (Available as: Abstract and PostScript)

Several communication abstractions were built upon Active Messages, including the Split-C shared address model, MPI message passing (described in http://now.CS.Berkeley.EDU/Fastcomm/MPI/performance/) and conventional sockets, described in the following:

- David E. Culler, Andrea Arpaci-Dusseau, Remzi Arpaci-Dusseau, Brent Chun, Steven Lumetta, Alan Mainwaring, Richard Martin, Chad Yoshikawa, Frederick Wong, "Parallel Computing on the Berkeley NOW," *JSPP'97 (9th Joint Symposium on Parallel Processing)* June 1997, Kobe, Japan. (Available as: Abstract and PostScript)

- Steve Rodrigues, Tom Anderson, David Culler, "High-Performance Local-Area Communication Using Fast Sockets," *USENIX '97*, 1997. (Available as: PostScript)

- Steve Rodrigues, "Building a Better Byte Stream," Master's Project Report, May 1996.

- Eric Anderson, David A. Patterson, "TheMagicrouter: An Application of Fast Packet Interposing," submitted to *OSDI '96*, 1996. (Available as: PostScript)

We addressed two fundamental questions in this work: how much communication performance could be obtained from cluster technology in general purpose use and how much do such improvements in communication performance improve the performance of real applications. We developed a novel technique for measuring the sensitivity for the basic components of communication performance, described in the following, and used it for evaluation on a range of parallel applications and traditional distributed system workloads. Overhead remains the most significant limiting factor.

- Richard Martin, Amin Vahdat, David Culler, Thomas Anderson, "Effects of Communication Latency, Overhead, and Bandwidth in a Cluster Architecture," *ISCA 24*, Denver, June 1997. (Available as: Abstract and PostScript)

- Richard Martin and David Culler, "NFS Sensitivity to High Performance Networks," *SIGMETRICS '99*, Atlanta, May 1999.

- Richard Martin, "Application Sensitivity to Network Performance," PhD Thesis, UC Berkeley, 1999.

Global Operating System Layer

We demonstrated a global layer operating system, Glunix, that scales beyond 100 workstations connected by low latency custom network interfaces via a high bandwidth switch. It provides a parallel network process abstraction and it able to run both parallel programs and unmodified sequential programs. This was a cornerstone in all research conducted on NOW and has executed many millions of jobs. A heartbeat monitor and automatic restart facility, GluGuard, was developed to maintain the collection of Glunix components across failures and incremental reconfigurations. The development of Glunix revealed several fundamental limitations in the classic Unix interface that compromise perfect virtualization. We developed a sophisticated interpositioning strategy to overcome these difficulties. Detail is provided in the following:

- Douglas Ghormley, "User Level Operating System Services, PhD Thesis, UC Berkeley, 1998.

- Douglas Ghormley, David Petrou, Steven Rodrigues, Amin Vahdat, Thomas Anderson, "GLUnix: A Global Layer Unix for a Network of Workstations," *Software-Practice and Experience*, Vol. 28, No. 9, July 25, 1998. (Available as: Abstract and PostScript)

One of the critical issues revealed in our technology evaluation phase was coordinated scheduling of parallel programs. Most programs are written with the assumption that the constituent processes actually run at the same time. Local operating system schedulers do not obey this discipline, and the lack of coscheduling can result in slowdowns of one or even two orders of magnitude relative to dedicated user. Although Glunix provided an explicit gang scheduling capability, that solution was unsatisfactory for several reasons. We wanted to be able to mix sequential and parallel jobs and allow interactive use. Gang scheduling is inefficient in presence of load imbalance or when constituent processes are operating independently. Its implementation is complex and introduces another set of potential failures and performance bottlenecks. The deep question was whether it was possible to design mechanisms where parallel programs could get themselves coscheduled over local schedulers when they require it-- using only the communication inherent to the program. We developed an elegant and simple adaptive technique, where the communication runtime observes how long it waits for responses and reacts by either continuing to wait or by blocking. A complete development and evaluation of the effectiveness of this approach over a wide range of applications and scenarios are provided by the following:

- Andrea Dusseau, Remzi Arpaci, David Culler, "Effective Distributed Scheduling of Parallel Workloads," *SIGMETRICS '96 Conference on Measurement and Modeling*, 1996. (Available as: PostScript)

- Andrea Arpaci-Dusseau, David Culler, Alan Mainwaring, "Scheduling with Implicit Information in Distributed Systems," *1998 SIGMETRICS Conference on the Measurement and Modeling of Computer Systems*, pp. 233-243, Madison, 24-26 June 1998. (Available as: Abstract and PostScript)

16

- Andrea Arpaci-Dusseau, David Culler "Extending Proportional-Share Scheduling to a Network of Workstations," *International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA '97)*, Las Vegas, June 1997. (Available as: Abstract and PostScript)

- Andrea Arpaci-Dusseau, "Implicit Coscheduling: Coordinated Scheduling with Implicit Information in Distributed Systems," Ph.D. thesis, UC Berkeley, December 1998. (Available as: Abstract and PostScript)

Scalable I/O

Our development of novel cluster-based file systems provided the basis for several deep analyses of the design of high-performance, fault tolerant file systems and the opportunities for organizing data and metadata across a network of storage devices. The following provide a complete design study of peer-to-peer or serverless file systems:

- Michael Dahlin, "Serverless Network File Systems," PhD thesis, UC Berkeley (1995). (Available as: PostScript)

- Tom Anderson, Michael Dahlin, Jeanna Neefe, David Patterson, Drew Roselli, Randy Wang Serverless Network File Systems, *IEEE TOCS (February 1996), 15th Symposium on Operating Systems Principles, ACM Transactions on Computer Systems*, 1995. (Available as: PostScript)

We completely reimplemented xFS following a "correct by construction" design methodology. This utilized a recent set of tools that allows protocols to be expressed in an application specific language which can be compiled either for a protocol verifier or a actual system. In addition, we developed a massive data tracing facility and used it to drive the design of novel storage managers. Finally, we developed a formal framework for optimizing the transfer of blocks on high-speed networks. Detail is provided in the following:

- Satish Chandra, Michael Dahlin, Bradley Richards, Randolph Wang, Thomas Anderson, James Larus, "Experience with a Language for Writing Coherence Protocols," To appear in USENIX Conference on Domain-Specific Languages *USENIX/DSL*, Santa Barbara, California, 15-17 October 1997. (Available as: Abstract and PostScript)

- Jeanna Neefe Matthews, Drew Roselli, Adam M. Costello, Randy Wang, Tom Anderson, "Improving the Performance of Log-Structured File Systems with Adaptive Methods," *SOSP 16* St. Malo, France, 5-8 October 1997. (Available as: Abstract and PostScript)

- Randolph Wang, Arvind Krishnamurthy, Richard Martin, Thomas Anderson, David Culler, "Modeling and Optimizing Pipeline Latency," 1998 SIGMETRICS Conference on the Measurement and Modeling of Computer Systems, Madison, Wisconsin, 24-26 June 1998. (Available as: Abstract and PostScript)

A fundamental strength of cluster architectures proved to be the ability to drive massive I/O bandwidth across a large number of independently attached disks. To explore this capability, we built a very high-performance parallel I/O facility and used it to set (and hold for two years) the world record disk-to-disk sorting benchmark (both the response oriented Datamation benchmark and the bandwidth oriented Minute Sort). Detail is in the following:

- Andrea Arpaci-Dusseau, Remzi Arpaci-Dusseau, David Culler, Joseph Hellerstein, David Patterson, "Searching for the Sorting Record: Experiences in Tuning NOW-Sort," *1998 Symposium on Parallel and Distributed Tools (SPDT '98)*, Welches, Oregon, 3-4 August 1998. (Available as: Abstract and PostScript)

- Andrea Arpaci-Dusseau, Remzi Arpaci-Dusseau, David Culler, Joseph Hellerstein, David Patterson, "High-Performance Sorting on Networks of Workstations," *SIGMOD '97*, Tucson, Arizona, May 1997. (Available as: Abstract and PostScript)

This investigation revealed that disks are a fundamental source of performance variation, due to mechanical characteristics, variations is transfer rates, and numerous other factors, which has largely been overlooked. Thus, we investigated the fundamental question of designing parallel I/O systems with robust performance and developed a novel adaptive scheme where applications are constructed as a composition of flows and computation is scheduled based on availability of data. This dataflow scheme generalizes techniques employed within DBMS systems. Replication provides a means of obtaining "performance availability," not just functional availability. Detail is provided in the following:

- Remzi Arpaci-Dusseau, Eric Anderson, Noah Treuhaft, David Culler, Joseph Hellerstein, David Patterson, Katherine Yelick, "Cluster I/O with River: Making the Fast Case Common," *IOPADS '99*. (Available as: Abstract, PostScript)

Cluster Architecture Design Space

As part of our evaluation of cluster architectures, we built additional large-scale clusters to understand the scope of the cluster design space.

Clusters of SMPs

One major dimension of this study was the basic node granularity and the design issues of multiple processors per node. We constructed a cluster of four 8-processor Enterprise 5000 SMPS, each with multiple network interfaces. Upon this system, we extended our Active Message system to utilize multiple network interfaces simultaneously and developed a multiprotocol version of Active Messages that transparently utilized shared memory within a node and the fast network between nodes. The hardware configuration was carefully chosen to permit a very controlled comparison of cluster and cluster-of-SMP architectures. Detail is provided in the following.

18

- Steven Lumetta, David Culler, "Managing Concurrent Access for Shared Memory Active Messages," To appear in *IPPS/SPDP 98*, Orlando, Florida, March 1998. (Available as: Abstract and PostScript)

- Steven Lumetta, Alan Mainwaring, David Culler, "Multi-Protocol Active Messages on a Cluster of SMPs," *SC'97*, San Jose, California, November 1997. (Available as: Abstract and PostScript)

- Steven Lumetta, "Design and Evaluation of Multi-Protocol Communication on a Cluster of SMPs," PhD thesis, UC Berkeley, November 1998.

Utilizing our high-performance I/O work, we conducted an in-depth architectural analysis across the spectrum from SMPs to Clusters.

- Remzi Arpaci-Dusseau, Andrea Arpaci-Dusseau, David Culler, Joseph Hellerstein, David Patterson, "The Architectural Costs of Streaming I/O: A Comparison of Workstations, Clusters, and SMPs," To appear in *HPCA 4*, Las Vegas, February 1998. (Available as: Abstract and PostScript)

Massive Storage Clusters

The second major dimension of the architectural investigation was the use of clustering to provide massive storage capability at a cost of a few percent over the raw media. In conjunction with the DARPA RoboLine grant and major donations from IBM and Intel, we built a massive storage cluster consisting of four terabytes of storage represented by several hundred 9 GB disks distributed over 24 PCs.

Tertiary Disk Massive Storage Cluster
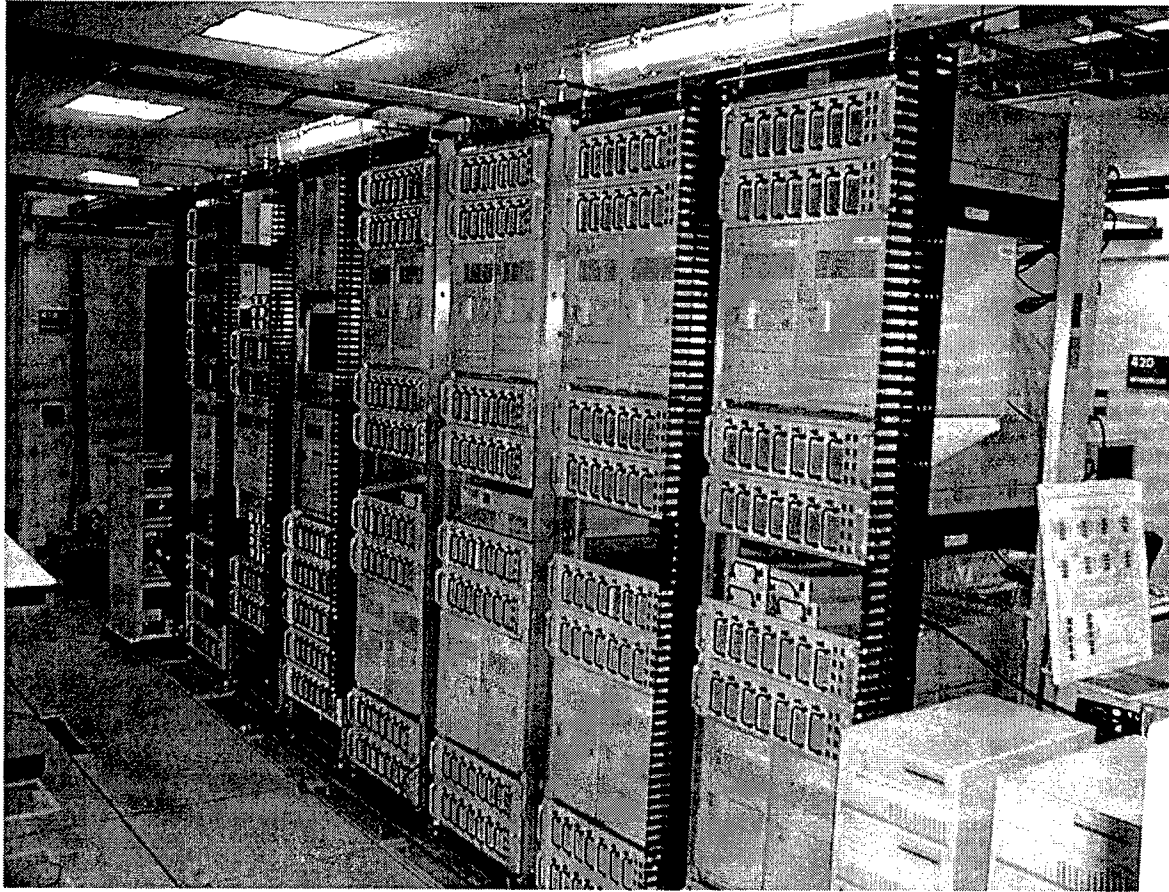
To drive this system, we used it to host the on-line collection of the San Francisco Museum of Fine Arts. Detail is in the following:

- Satoshi Asami, Nisha Talagala, Thomas Anderson, Ken Lutz, David Patterson, "The Design of Large-Scale, Do-It-Yourself RAIDs," *White Paper*, 1995. (Available as: PostScript)

- Randy Katz, Thomas Anderson, John Ousterhout, David Patterson, "Robo-line Storage: Low Latency, High Capacity Storage Systems Over Geographically Distributed Networks," *White Paper*, 1991. (Available as: PostScript)

In conjunction with the Spin project at University of Washington, we also explored the utility of off-loading protocol logic into the network interface.

- Marc Fiuczynski , Richard Martin, Tsutomu Owa, Brian Bershad, "On Using Intelligent Network Interface Cards to support Multimedia Applications," *8th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 98)*, Cambridge, UK, July 1998. (Available as: PostScript)

Scalable, Available Internet Services

A very large number of demanding applications were developed and evaluated on NOW, spanning a rich spectrum of areas from architectural simulation, machine learning, protocol verification, dynamic solids modeling, numerical linear algebra, adaptive mesh refinement, finite element modeling, maximum likelihood genetics, parallel rendering, improcessing, compression, real-time video effects, content distillation, latent semantic indexing, and several others. However, the massive impact was realized in the form of scalable internet services, which exploiting the availability, I/O capacity and bandwidth, and fast communication of the cluster. The most dramatic of these was the Inktomi search engine, originally prototyped on NOW and rapidly transitioned into industry. Today the Inktomi clusters serve roughly 50% of the searches on the web, servicing 20 million distinct users through numerous popular search interfaces and portals. It is still running a version of Active Messages derived directly from the Technology Exporation phase of the NOW project. Detail of the design is given in the following:

- Eric Brewer, "Delivering High Availability for Inktomi Search Engines," *SIGMOD Record ACM Special Interest Group on Management of Data*, vol. 27 no. 2, 1998.
- Eric Brewer, "Clusters: Multiply and Conquer," *Data Communications* July 1997.

We investigated the larger research question of extending core aspects of cluster technology out to the wide area. One of the key discoveries was the use of moving functionality traditionally associated with "front-end" machines, such as load-balancing and fail-over, automatically into the client. A second was the extension of the cache consistent network file system and OS interpositioning for service replication to remote clusters. Detail is in the following:

- Chad Yoshikawa, Brent Chun, Paul Eastham, Amin Vahdat, Tom Anderson, David Culler, "Using Smart Clients To Build Scalable Services," *USENIX '97*, 1997. (Available as: PostScript)

- Amin Vahdat, Tom Anderson, Mike Dahlin, Eshwar Belani, David Culler, Paul Eastham, and Chad Yoshikawa, "WebOS: Operating System Services For Wide Area Applications," *Seventh Symposium on High Performance Distributed Computing*, July 1998.(Available as: PostScript)

- Amin Vahdat, "Operating System Services for Wide-Area Applications," PhD thesis, UC Berkeley, December 1998.

A second fundamental area of development was the use of scalable transcoding services in the cluster to support constrained, poorly connected, or mobile devices. This investigation was done jointly with two other DARPA projects. The first prototype was the Transcend proxy, developed as part of the BARWAN project, to provide on-the-fly content distillation to deliver content to small clients over bandwidth constrained links.

- Fox, I. Goldberg, Steven Gribble, D. Lee, A. Polito, and Eric Brewer, "Experience with TopGun Wingman: A Proxy-Based Web Browser for the 3Com PalmPilot," *Proceedings of Middleware '98*, Lake District, England, September 1998. (Available as: PostScript)

The second was the Media Gateway proxy developed as part of the MASH project. In this case, the proxy participates as a well-connected part of a multicast-based video session, but it down samples the multimedia stream to match the limited bandwidth and functionality of a client, which may not even be able to participate in multicast.

- Elan Amir, Steven McCanne, and Randy Katz, "An Active Service Framework and its Application to Real-time Multimedia Transcoding," *Proceedings of ACM SIGCOMM '98*, Vancouver, British Columbia, September 1998.

For both of these services, a portion of the NOW cluster was set aside for baseline capability and has the load increased the proxy service would negotiate with the Glunix layer to obtain additional transient resources.

Although the main goal of DARPA funding is making high-risk, high-reward research a reality, it also produces outstanding graduates. Below are the names, final degrees, and first jobs of alumni who worked on NOW as graduate students:

Name	Degree	First Institution	Title
Andrea Arpaci-Dusseau	Ph.D.	University of Wisconsin, Computer Science Division	Assistant Professor
Remzi Arpaci-Dusseau	Ph.D.	University of Wisconsin, Computer Science Division	Assistant Professor
Michael Dahlin	Ph.D.	University of Texas, Austin Computer Science Division	Assistant Professor
Steven Lumetta	Ph.D.	University of Illinois Computer Science Division	Assistant Professor
Richard Martin	Ph.D.	Rutgers University Computer Science Division	Assistant Professor
Amin Vahdat	Ph.D.	Duke University Computer Science Division	Assistant Professor
Randy Wang	Ph.D.	Princeton University Computer Science Division	Assistant Professor

DISTRIBUTION LIST

addresses	number of copies
CRAIG S. ANKEN AFRL/IFTB 525 BROOKS ROAD ROME, NY 13441-4505	5
PROF. DAVID E. CULLER UNIVERSITY OF CALIFORNIA, BERKELEY DEPT OF COMPUTER SCIENCE 627 SODA HALL #1776 BERKELEY, CA 94720-1776	1
AFRL/IFOIL TECHNICAL LIBRARY 26 ELECTRONIC PKY ROME NY 13441-4514	1
ATTENTION: DTIC-OCC DEFENSE TECHNICAL INFO CENTER 3725 JOHN J. KINGMAN ROAD, STE 0944 FT. BELVOIR, VA 22060-6218	2
DEFENSE ADVANCED RESEARCH PROJECTS AGENCY 3701 NORTH FAIRFAX DRIVE ARLINGTON VA 22203-1714	1
ATTN: NAN PFRIMMER IIT RESEARCH INSTITUTE 201 MILL ST. ROME, NY 13440	1
AFIT ACADEMIC LIBRARY AFIT/LDR, 2950 P.STREET AREA B, BLDG 642 WRIGHT-PATTERSON AFB OH 45433-7765	1
AFRL/HESC-TDC 2698 G STREET, BLDG 190 WRIGHT-PATTERSON AFB OH 45433-7604	1

```
ATTN:   SMDC IM PL                              1
US ARMY SPACE & MISSILE DEF CMD
P.O. BOX 1500
HUNTSVILLE AL 35807-3801


COMMANDER, CODE 4TL000D                         1
TECHNICAL LIBRARY, NAWC-WD
1 ADMINISTRATION CIRCLE
CHINA LAKE  CA  93555-6100


CDR, US ARMY AVIATION & MISSILE CMD             2
REDSTONE SCIENTIFIC INFORMATION CTR
ATTN: AMSAM-RD-OB-R, (DOCUMENTS)
REDSTONE ARSENAL AL 35898-5000


REPORT LIBRARY                                  1
MS P364
LOS ALAMOS NATIONAL LABORATORY
LOS ALAMOS NM 87545


ATTN:   D'BORAH HART                            1
AVIATION BRANCH SVC 122.10
FOB10A, RM 931
800 INDEPENDENCE AVE, SW
WASHINGTON DC  20591

AFIWC/MSY                                       1
102 HALL BLVD, STE 315
SAN ANTONIO TX 78243-7016


ATTN:   KAROLA M. YOURISON                      1
SOFTWARE ENGINEERING INSTITUTE
4500 FIFTH AVENUE
PITTSBURGH PA 15213


USAF/AIR FORCE RESEARCH LABORATORY              1
AFRL/VSOSA(LIBRARY-BLDG 1103)
5 WRIGHT DRIVE
HANSCOM AFB  MA  01731-3004


ATTN:   EILEEN LADUKE/D460                      1
MITRE CORPORATION
202 BURLINGTON RD
BEDFORD MA 01730
```

```
OUSD(P)/DTSA/DUTD                               1
ATTN:  PATRICK G. SULLIVAN, JR.
400 ARMY NAVY DRIVE
SUITE 300
ARLINGTON VA 22202

SOFTWARE ENGR'G INST TECH LIBRARY               1
ATTN:  MR DENNIS SMITH
CARNEGIE MELLON UNIVERSITY
PITTSBURGH PA 15213-3890


USC-ISI                                         1
ATTN:  DR ROBERT M. BALZER
4676 ADMIRALTY WAY
MARINA DEL REY CA 90292-6695


KESTREL INSTITUTE                               1
ATTN:  DR CORDELL GREEN
1801 PAGE MILL ROAD
PALO ALTO CA 94304


ROCHESTER INSTITUTE OF TECHNOLOGY               1
ATTN:  PROF J. A. LASKY
1 LOMB MEMORIAL DRIVE
P.O. BOX 9887
ROCHESTER NY 14613-5700

AFIT/ENG                                        1
ATTN:TOM HARTRUM
WPAFB OH 45433-6583


THE MITRE CORPORATION                           1
ATTN:  MR EDWARD H. BENSLEY
BURLINGTON RD/MAIL STOP A350
BEDFORD MA 01730


ANDREW A. CHIEN                                 1
SAIC CHAIR PROF (SCI APL INT CORP)
USCD/CSE-AP&M 4808
9500 GILMAN DRIVE, DEPT. 0114
LAJOLLA CA 92093-0114

HONEYWELL, INC.                                 1
ATTN:  MR BERT HARRIS
FEDERAL SYSTEMS
7900 WESTPARK DRIVE
MCLEAN VA 22102
```

```
SOFTWARE ENGINEERING INSTITUTE                      1
ATTN:  MR WILLIAM E. HEFLEY
CARNEGIE-MELLON UNIVERSITY
SEI 2218
PITTSBURGH PA 15213-38990


UNIVERSITY OF SOUTHERN CALIFORNIA                   1
ATTN:  DR. YIGAL ARENS
INFORMATION SCIENCES INSTITUTE
4676 ADMIRALTY WAY/SUITE 1001
MARINA DEL REY CA 90292-6695


COLUMBIA UNIV/DEPT COMPUTER SCIENCE                 1
ATTN:  DR GAIL E. KAISER
450 COMPUTER SCIENCE BLDG
500 WEST 120TH STREET
NEW YORK NY 10027


AFIT/ENG                                            1
ATTN:  DR GARY B. LAMONT
SCHOOL OF ENGINEERING
DEPT ELECTRICAL & COMPUTER ENGRG
WPAFB OH 45433-6583


NSA/OFC OF RESEARCH                                 1
ATTN:  MS MARY ANNE OVERMAN
9800 SAVAGE ROAD
FT GEORGE G. MEADE MD 20755-6000


AT&T BELL LABORATORIES                              1
ATTN:  MR PETER G. SELFRIDGE
ROOM 3C-441
600 MOUNTAIN AVE
MURRAY HILL NJ 07974


ODYSSEY RESEARCH ASSOCIATES, INC.                   1
ATTN:  MS MAUREEN STILLMAN
3014 HARRIS B. DATES DRIVE
ITHACA NY 14850-1313


TEXAS INSTRUMENTS INCORPORATED                      1
ATTN:  DR DAVID L. WELLS
P.O. BOX 655474, MS 238
DALLAS TX 75265


KESTREL DEVELOPMENT CORPORATION                     1
ATTN:  DR RICHARD JULLIG
3260 HILLVIEW AVENUE
PALO ALTO CA 94304
```

DARPA/ITO   1
ATTN: DR KIRSTIE BELLMAN
3701 N FAIRFAX DRIVE
ARLINGTON VA 22203-1714

NASA/JOHNSON SPACE CENTER   1
ATTN: CHRIS CULBERT
MAIL CODE PT4
HOUSTON TX 77058

STERLING IMD INC.   1
KSC OPERATIONS
ATTN: MARK MAGINN
BEECHES TECHNICAL CAMPUS/RT 26 N.
ROME NY 13440

HUGHES SPACE & COMMUNICATIONS   1
ATTN: GERRY BARKSDALE
P. O. BOX 92919
BLDG R11 MS M352
LOS ANGELES, CA 90009-2919

SCHLUMBERGER LABORATORY FOR   1
   COMPUTER SCIENCE
ATTN: DR. GUILLERMO ARANGO
8311 NORTH FM620
AUSTIN, TX 78720

DECISION SYSTEMS DEPARTMENT   1
ATTN: PROF WALT SCACCHI
SCHOOL OF BUSINESS
UNIVERSITY OF SOUTHERN CALIFORNIA
LOS ANGELES, CA 90089-1421

SOUTHWEST RESEARCH INSTITUTE   1
ATTN: BRUCE REYNOLDS
6220 CULEBRA ROAD
SAN ANTONIO, TX 78228-0510

NATIONAL INSTITUTE OF STANDARDS   1
   AND TECHNOLOGY
ATTN: CHRIS DABROWSKI
ROOM A266, BLDG 225
GAITHSBURG MD 20899

EXPERT SYSTEMS LABORATORY   1
ATTN: STEVEN H. SCHWARTZ
NYNEX SCIENCE & TECHNOLOGY
500 WESTCHESTER AVENUE
WHITE PLAINS NY 20604

```
NAVAL TRAINING SYSTEMS CENTER                          1
ATTN:  ROBERT BREAUX/CODE 252
12350 RESEARCH PARKWAY
ORLANDO FL 32826-3224


DR JOHN SALASIN                                        1
DARPA/ITO
3701 NORTH FAIRFAX DRIVE
ARLINGTON VA 22203-1714


DR BARRY BOEHM                                         1
DIR, USC CENTER FOR SW ENGINEERING
COMPUTER SCIENCE DEPT
UNIV OF SOUTHERN CALIFORNIA
LOS ANGELES CA 90089-0781


DR STEVE CROSS                                         1
CARNEGIE MELLON UNIVERSITY
SCHOOL OF COMPUTER SCIENCE
PITTSBURGH PA 15213-3891


DR MARK MAYBURY                                        1
MITRE CORPORATION
ADVANCED INFO SYS TECH: G041
BURLINTON ROAD, M/S K-329
BEDFORD MA 01730

ISX                                                    1
ATTN:  MR. SCOTT FOUSE
4353 PARK TERRACE DRIVE
WESTLAKE VILLAGE,CA 91361


MR GARY EDWARDS                                        1
ISX
433 PARK TERRACE DRIVE
WESTLAKE VILLAGE CA 91361


DR ED WALKER                                           1
BBN SYSTEMS & TECH CORPORATION
10 MOULTON STREET
CAMBRIDGE MA 02238


LEE ERMAN                                              1
CIMFLEX TEKNOWLEDGE
1810 EMBACADERO ROAD
P.O. BOX 10119
PALO ALTO CA 94303
```

DR. DAVE GUNNING   1
DARPA/ISO
3701 NORTH FAIRFAX DRIVE
ARLINGTON VA   22203-1714

DAN WELD   1
UNIVERSITY OF WASHINGTON
DEPART OF COMPUTER SCIENCE & ENGIN
BOX 352350
SEATTLE, WA 98195-2350

STEPHEN SODERLAND   1
UNIVERSITY OF WASHINGTON
DEPT OF COMPUTER SCIENCE & ENGIN
BOX 352350
SEATTLE, WA 98195-2350

DR. MICHAEL PITTARELLI   1
COMPUTER SCIENCE DEPART
SUNY INST OF TECH AT UTICA/ROME
P.O. BOX 3050
UTICA, NY 13504-3050

CAPRARO TECHNOLOGIES, INC   1
ATTN: GERARD CAPRARO
311 TURNER ST.
UTICA, NY 13501

USC/ISI   1
ATTN: BOB MCGREGOR
4676 ADMIRALTY WAY
MARINA DEL REY, CA 90292

SRI INTERNATIONAL   1
ATTN: ENRIQUE RUSPINI
333 RAVENSWOOD AVE
MENLO PARK, CA 94025

DARTMOUTH COLLEGE   1
ATTN: DANIELA RUS
DEPT OF COMPUTER SCIENCE
11 ROPE FERRY ROAD
HANOVER, NH 03755-3510

UNIVERSITY OF FLORIDA   1
ATTN: ERIC HANSON
CISE DEPT 456 CSE
GAINESVILLE, FL 32611-6120

```
CARNEGIE MELLON UNIVERSITY                          1
ATTN:  TOM MITCHELL
COMPUTER SCIENCE DEPARTMENT
PITTSBURGH, PA 15213-3890


CARNEGIE MELLON UNIVERSITY                          1
ATTN:  MARK CRAVEN
COMPUTER SCIENCE DEPARTMAN
PITTSBURGH, PA 15213-3890


UNIVERSITY OF ROCHESTER                             1
ATTN:  JAMES ALLEN
DEPARTMENT OF COMPUTER SCIENCE
ROCHESTER, NY 14627


TEXTWISE, LLC                                       1
ATTN:  LIZ LIDDY
2-121 CENTER FOR SCIENCE & TECH
SYRACUSE, NY 13244


WRIGHT STATE UNIVERSITY                             1
ATTN:  DR. BRUCE BERRA
DEPART OF COMPUTER SCIENCE & ENGIN
DAYTON, OHIO 45435-0001


UNIVERSITY OF FLORIDA                               1
ATTN:  SHARMA CHAKRAVARTHY
COMPUTER & INFOR SCIENCE DEPART
GAINESVILLE, FL 32622-6125


KESTREL INSTITUTE                                   1
ATTN:  DAVID ESPINOSA
3260 HILLVIEW AVENUE
PALO ALTO, CA 94304


USC/INFORMATION SCIENCE INSTITUTE                   1
ATTN:  DR. CARL KESSELMAN
11474 ADMIRALTY WAY, SUITE 1001
MARINA DEL REY, CA 90292


MASSACHUSETTS INSTITUTE OF TECH                     1
ATTN:  DR. MICHAELE SIEGEL
SLOAN SCHOOL
77 MASSACHUSETTS AVENUE
CAMBRIDGE, MA 02139
```

```
USC/INFORMATION SCIENCE INSTITUTE                    1
ATTN:  DR. WILLIAM SWARTHOUT
11474 ADMIRALTY WAY, SUITE 1001
MARINA DEL REY, CA 90292


STANFORD UNIVERSITY                                  1
ATTN:  DR. GIO WIEDERHOLD
857 SIERRA STREET
STANFORD
SANTA CLARA COUNTY, CA 94305-4125

NCCOSC RDTE DIV D44208                               1
ATTN:  LEAH WONG
53245 PATTERSON ROAD
SAN DIEGO, CA 92152-7151


SPAWAR SYSTEM CENTER                                 1
ATTN:  LES ANDERSON
271 CATALINA BLVD, CODE 413
SAN DIEGO CA 92151


GEORGE MASON UNIVERSITY                              1
ATTN:  SUSHIL JAJODIA
ISSE DEPT
FAIRFAX, VA 22030-4444


DIRNSA                                               1
ATTN:  MICHAEL R. WARE
DOD, NSA/CSS (R23)
FT. GEORGE G. MEADE MD 20755-6000


DR. JIM RICHARDSON                                   1
3660 TECHNOLOGY DRIVE
MINNEAPOLIS, MN 55418


LOUISIANA STATE UNIVERSITY                           1
COMPUTER SCIENCE DEPT
ATTN:  DR. PETER CHEN
257 COATES HALL
BATON ROUGE, LA 70803

INSTITUTE OF TECH DEPT OF COMP SCI                   1
ATTN:  DR. JAIDEEP SRIVASTAVA
4-192 EE/CS
200 UNION ST SE
MINNEAPOLIS, MN 55455
```

```
GTE/BBN                                              1
ATTN:   MAURICE M. MCNEIL
9655 GRANITE RIDGE DRIVE
SUITE 245
SAN DIEGO, CA 92123

UNIVERSITY OF FLORIDA                                1
ATTN:   DR. SHARMA CHAKRAVARTHY
E470 CSE BUILDING
GAINESVILLE, FL 32611-6125


AFRL/IFT                                             1
525 BROOKS ROAD
ROME, NY 13441-4505


AFRL/IFTM                                            1
525 BROOKS ROAD
ROME, NY 13441-4505
```

MISSION
OF
AFRL/INFORMATION DIRECTORATE (IF)

The advancement and application of information systems science and

technology for aerospace command and control and its transition to air,

space, and ground systems to meet customer needs in the areas of Global

Awareness, Dynamic Planning and Execution, and Global Information

Exchange is the focus of this AFRL organization. The directorate's areas

of investigation include a broad spectrum of information and fusion,

communication, collaborative environment and modeling and simulation,

defensive information warfare, and intelligent information systems

technologies.